МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

Кафедра «Информационные технологии и компьютерные системы»

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №10

по дисциплине

«Проектирование баз данных»

Вариант 8

Выполнил:

Донец Н.О.

Проверил:

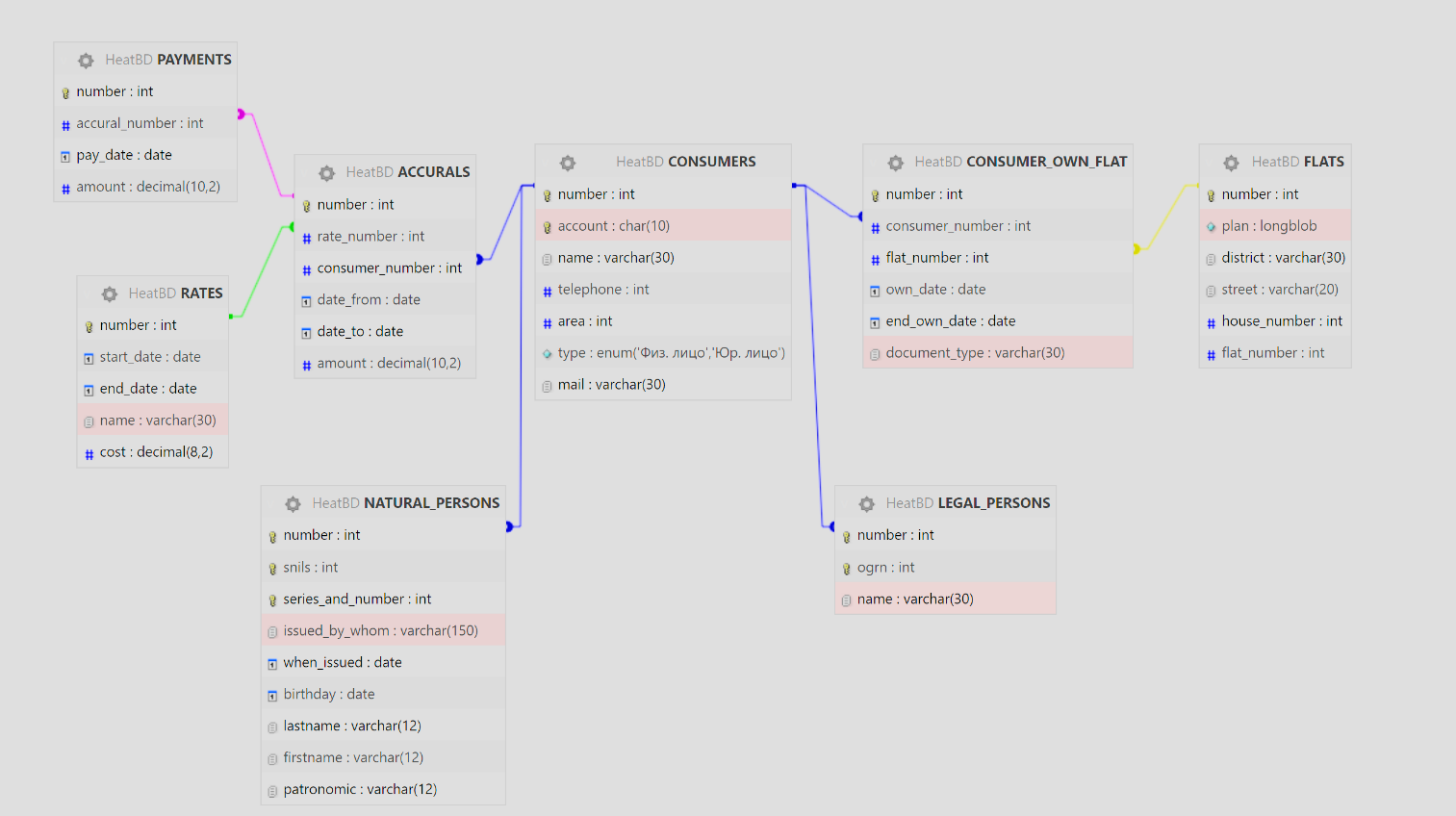
Волкова Т. В.

Севастополь

2024 г.

**Задание**

Разработать БД для ежемесячного учета оплаты услуг за теплоснабжение в теплосети района. БД должна содержать следующие данные: адрес (район, улица, номера дома и квартиры), план и площадь квартиры, месячные тарифы за потребленное тепло из расчета за 1 квадратный метр площади, общая плата за услуги, сведения об уплате с указанием дат и сумм, показатели оплаты услуг за отчетное полугодие по домам.



**5. ТЕСТИРОВАНИЕ БАЗЫ ДАННЫХ**

**5.5 SQL-запросы на создание пользовательских представлений**

- Для базы данных были сформированы и выполнены следующие запросы:

1) SQL-запрос на создание представления для ввода данных в таблицу (с использованием ограничений на ввод) (Рисунок 1);

CREATE VIEW EXPENSIVE\_RATES AS

SELECT number, name, cost

FROM RATES WHERE cost > 4000

WITH CHECK OPTION;

2) SQL-запрос, который продемонстрирует успешный ввод данных через созданное представление (с учетом заданных ограничений на ввод) (Рисунок 2);

INSERT INTO expensive\_rates (number, name, cost) VALUES (7, 'Дорогой', 5000);

3) SQL-запрос, который продемонстрирует отрицательный результат при вводе данных через созданное представление (с учетом заданных ограничений на ввод) (Рисунок 3);

INSERT INTO expensive\_rates (number, name, cost) VALUES (7, 'Не Дорогой', 3500);

4) SQL-запрос на удаление созданного представления;

DROP VIEW expensive\_rates;

5) SQL-запрос на создание представления для вывода неполных данных из двух или более таблиц одновременно на основе заданного условия (Рисунок 4).

CREATE VIEW ACCURALS\_AND\_RATES AS

SELECT ACCURALS.number AS 'accural number', amount AS 'for all' , cost AS 'for 1 m', RATES.number AS 'rate number'

FROM ACCURALS JOIN RATES ON ACCURALS.rate\_number = RATES.number;



Рисунок 1 – Представление expensive\_rates

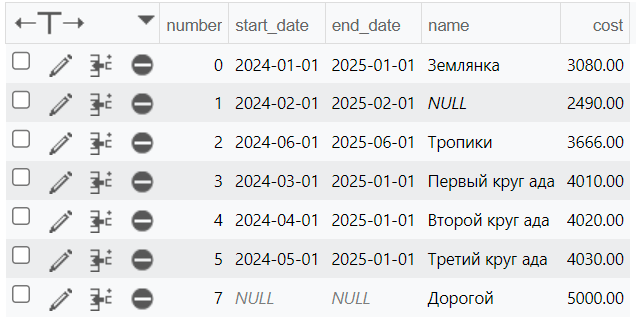


Рисунок 2 – Ввод данных в представление

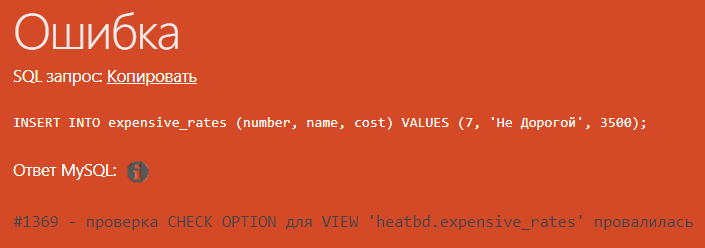


Рисунок 3 – Безуспешный ввод данных в представление

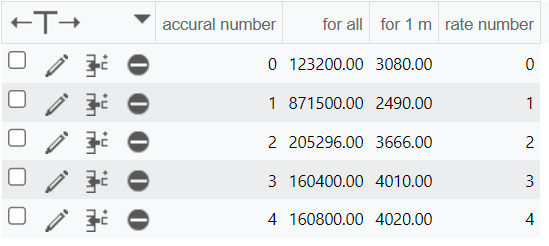


Рисунок 4 – Представление для вывода данных

**5.6. SQL-запросы на создание индексов**

- Были разработаны и выполнены следующие запросы:

1. SQL-запрос на создание уникального индекса, выборку и удаление индекса (Рисунок 5);

CREATE UNIQUE INDEX ind\_name ON RATES(name);

SELECT \* FROM RATES WHERE name = 'Землянка';

DROP INDEX ind\_name ON RATES

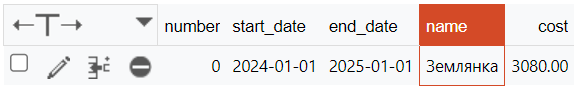


Рисунок 5 – Выборка уникального индекса

1. SQL-запрос на создание индекса на поле, по которому сортируются данные, выборку и удаление индекса (Рисунок 6);

CREATE INDEX name\_cost ON RATES(name, cost);

SELECT \* FROM RATES WHERE NAME IS NOT NULL ORDER BY cost;

DROP INDEX name\_cost ON RATES;

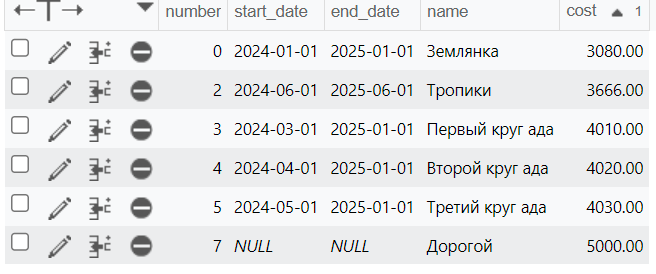


Рисунок 6 – Выборка индекса для поля, по которому сортируются данные

1. SQL-запрос на создание индекса на поле, по которому происходит группировка во время агрегации, выборку и удаление индекса (Рисунок 7);

CREATE INDEX sdate ON RATES(start\_date);

[SELECT](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) start\_date, [AVG](http://127.0.0.1/openserver/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(cost) FROM RATES GROUP BY start\_date;

DROP INDEX sdate ON RATES;

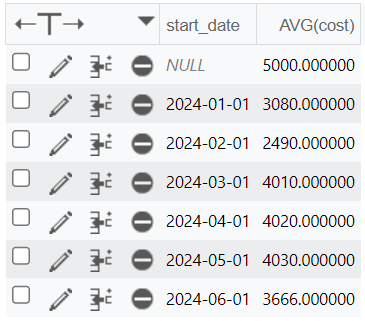


Рисунок 7 – Выборка индекса по которому происходит группировка во время агрегации

1. SQL-запрос на создание составного индекса с учетом селективности колонок, выборку и удаление индекса (Рисунок 8);

CREATE INDEX id\_name ON RATES(number, name);

SELECT \* FROM RATES WHERE number > 3 AND name IS NOT NULL;

DROP INDEX id\_name ON RATES;

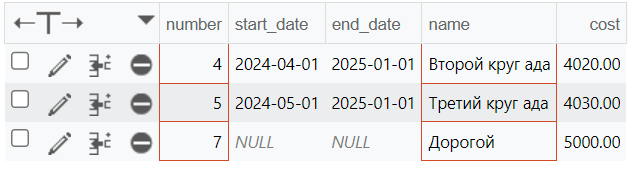


Рисунок 8 – Выборка индекса с учетом селективности колонок

Для индексации были выбраны поля number и name, т.к. number – это первичный ключ и он является уникальным, а name – название тарифа, которое имеет практически нулевую вероятность повторения.

1. SQL-запрос на создание составного индекса, выборку и удаление индекса (Рисунок 9).

CREATE INDEX id\_name\_c ON RATES(number, name, cost);

SELECT \* FROM RATES WHERE number > 3 AND name IS NOT NULL ORDER BY cost;

DROP INDEX id\_name\_c ON RATES;

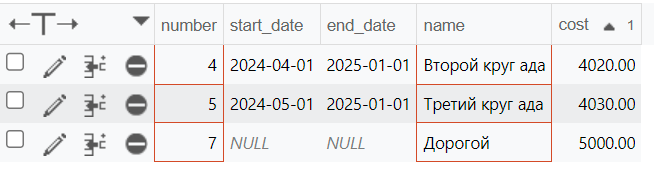


Рисунок 9 – Выборка составного индекса

На первое место в индексе было поставлено поле number, т.к. оно будет первым в запросе и является первичным ключом, на последнее место было поставлено поле cost, т.к. по нему проводится сортировка

**Вывод**

В ходе практической работы было проведено тестирование базы данных. Были написаны запросы на создание всевозможных представлений с одним или несколькими источниками данных, представления для ввода и вывода данных. Также были созданы индексы как одиночных, так и составных.